Polarity and the Logic of Delimited Continuations

Noam Zeilberger
Université Paris VII
noam@pps.jussieu.fr

Abstract—**Polarized logic is the logic of values and continuations, and their interaction through continuation-passing style. The main limitations of this logic are the limitations of CPS: that continuations cannot be composed, and that programs are fully sequentialized. Delimited control operators were invented in response to the limitations of classical continuation-passing. That suggests the question: what is the logic of delimited continuations?**

We offer a simple account of delimited control, through a natural generalization of the classical notion of polarity. This amounts to breaking the perfect symmetry between positive and negative polarity in the following way: answer types are positive. Despite this asymmetry, we retain all of the classical polarized connectives, and can explain "intuitionistic polarity" (e.g., in systems like CBPV) as a restriction on the use of connectives, i.e., as a logical fragment. Our analysis complements and generalizes existing accounts of delimited control operators, while giving us a rich logical language through which to understand the interaction of control with monadic effects.

I. Introduction

The original motivation for studying *polarity* as a property of logical connectives was to decompose the classical double-negation translations, and so to better understand the different ways in which classical logic could be endowed with constructive content [18], [19]. Polarity was a natural distinction in the context of linear logic (positive $\otimes$, $\oplus$, $\exists$ vs. negative $⅋$, $\&$, $\forall$), but once this classification was discovered, it became clear that polarity in linear logic was an elegant demonstration of a much more general phenomenon. Classical logic could be understood constructively by way of embedding into linear logic [8], [29], but it could also be understood on its own just by *polarizing* the connectives, i.e., by forcing them to have positive or negative polarity. This was to some extent already explicit in Girard's original papers, and is implicit in other constructive interpretations of classical logic, such as Selinger's (co)control categories [41].

Formally, applying Andreoli's notion of *focusing* [3] to classical logic gives a canonical representation of proofs which are β-normal, η-long, and in *continuation-passing style.* The particular fragment of CPS is determined by how one polarizes the classical connectives. Polarized logic—where the polarity of connectives is made explicit, and connectives come in both positive and negative varieties—can in this sense be seen as a logic for expressing evaluation order at the local level of proofs and types, rather than as a global property of a language (cf. [47]).

Two questions arise naturally.

First, one does not need to start with classical (linear or non-linear) logic to study polarization and focalization. Focusing has been applied successfully to guide proof search in intuitionistic logic [23], [6], [33], while distinctions seemingly related to classical polarity have arisen in intuitionistic type-theoretic settings. Notably, Filinski [15] and Levy [30] distinguish *value* types and *computation* types to decompose the monadic view of effects in functional programming, while Watkins et al. [45] distinguish *synchronous* and *asynchronous* types (adopting Andreoli's terminology for positive and negative polarity in classical linear logic) to tame equality in a dependent logical framework. Yet, intuitionistic logic seems to blur the precise duality between positive and negative polarity, and to preclude certain polarizations of the connectives—for example, none of these systems includes a negative polarity disjunction. Moreover, there doesn't appear to be the same connection with double-negation translation or CPS, although there is some connection to sequentiality (cf. [45], [12], [46]). In short, it is unclear exactly how the phenomenon of polarity in intuitionistic logic is related to the classical case.

Second, from a practical and theoretical perspective, classical polarity may not give a sufficiently refined notion of evaluation order, precisely *because* of this correspondence with classical continuation-passing. The CPS discipline is a kind of paranoia, where a term needs to have its entire life planned before it takes a step. This means that a program ends up completely sequentialized even in cases where order of evaluation is irrelevant. Historically, this was the one of the reasons for the invention of "composable" or "delimited" control operators [13], [10], [11], which allow a program to mix continuation-passing and direct style. Besides numerous practical applications (see the long bibliography of Kiselyov and Shan [28]), one hint of the foundational relevance of delimited control is a tantalizing representation theorem due to Filinski [14], which states that the operators *shift* and *reset* can be used to perform *monadic reflection,* i.e., to mediate between internal and external views of computational effects.

Can classical and intuitionistic flavors of polarity be reconciled, and can the relationship between polarized logic and CPS be extended to delimited continuations?

In this paper, we describe a natural generalization of classical polarized logic, and explain how it accounts both for intuitionistic polarity and for delimited control. Our basic premise is that the perfect symmetry between positive and negative polarity should be broken: *answer types* (i.e., continuations' result types) are positive. Starting from this thesis, we generalize the proof theory of polarity in a more or less directed way, with the important difference that the notion of equality on proofs may now incorporate the equations of any given monad, relaxing the strict sequentiality of CPS. Perhaps surprisingly, we can retain (indeed generalize) all of the usual polarized connectives (e.g., $\parr$), viewing traditional intuitionistic systems as *logical fragments*—in effect giving a double-negation interpretation *of* intuitionistic logic. We apply our logical analysis of delimited control operators first to explain Danvy and Filinski's original type system, then to revisit Filinski's representation theorem, showing its elementary character and close link to the Yoneda lemma.

II. Classical polarity and control

Before we go on to describe this generalization of classical polarized logic and its relationship to intuitionistic polarity and delimited control, we review the correspondence between classical polarity and undelimited control. Although the connection between polarities and CPS is by now well-established, it is not always well understood, sometimes thought to be an exotic property of linear logic. The connection is mundane, and can be understood with the intuitions of ordinary functional programming.

Positive and negative polarity are simply two different ways of defining types. A positive type is defined by describing the shape of its values—in other words, like any datatype. If this is taken as a complete definition, i.e., if these are *all* the possible shapes of a value of positive type, then we are justified in using ordinary pattern-matching notation to define the continuations for that type. Dually, a negative type is defined by describing the shape of its continuations—similar to how one defines a record or class, by listing its fields/methods. Again, if this is taken as a complete definition, then a value of negative type can be constructed by *pattern-matching on continuations.* Values of negative type are therefore understood as control operators, and naturally *lazy*—they can be computed on demand in response to their continuation. Indeed, the ubiquitous lambda of ordinary functional programming can be understood in this way, as a degenerate kind of control operator: defined by matching on a continuation consisting of a value for the argument, and a continuation for the result.

The subtlety in this description arises at the interaction of positive and negative polarity. A positive value may contain negative subcomponents. For a continuation accepting positive type, this is the point where pattern-matching must end by binding a value variable, because nothing can be assumed about the shape of negative values. Again, this is an everyday fact of life in ML or Haskell, a reflection of the impossibility of pattern-matching against functional values except by binding a variable. Likewise, a negative continuation can contain positive subcomponents. For a negative value, this is the point where pattern-matching on continuations must stop, binding the positive continuation to a continuation variable.

A. Proofs

We use logic to make these intuitions precise. Figure 1 presents the rules for building focusing proofs in polarized classical logic. The presentation is a bit unconventional, so we'll explain it gradually (for more background see [47]). We

Contexts $\Delta, \Gamma \ ::= \ \cdot \mid \Delta_1, \Delta_2 \mid N \mid \bullet P$

$$\frac{\Delta \Vdash [P] \quad \Gamma \vdash \Delta}{\Gamma \vdash [P]} \qquad \frac{\Delta \Vdash [P] \quad \longrightarrow \quad \Gamma, \Delta \vdash \#}{\Gamma \vdash \bullet P}$$

$$\frac{\Delta \Vdash [\bullet N] \quad \longrightarrow \quad \Gamma, \Delta \vdash \#}{\Gamma \vdash N} \qquad \frac{\Delta \Vdash [\bullet N] \quad \Gamma \vdash \Delta}{\Gamma \vdash [\bullet N]}$$

$$\frac{N \in \Gamma \quad \Gamma \vdash [\bullet N]}{\Gamma \vdash \#} \qquad \frac{\bullet P \in \Gamma \quad \Gamma \vdash [P]}{\Gamma \vdash \#}$$

$$\frac{}{\Gamma \vdash \cdot} \qquad \frac{\Gamma \vdash \Delta_1 \quad \Gamma \vdash \Delta_2}{\Gamma \vdash \Delta_1, \Delta_2}$$

Fig. 1. The classical rules of focusing

use the letters P and N to range over positive and negative propositions, respectively. There are two basic stances we can take about a proposition: assertion or denial. For positive propositions, we write these respective judgments as $[P]$ and $\bullet P$, while for negative propositions we write N and $[\bullet N]$. As we will explain shortly, assertion and denial are merely another way of reading positionality in "multiple conclusions" sequent calculus (cf. [38]), while the brackets mark *focus* in the sense of Andreoli [3].

Simple contexts Δ, Γ are formed by combining (using the comma) assertions N of negative propositions and denials $\bullet P$ of positive propositions, treated up to associativity and unit laws for the empty context ($\cdot$). We omit hypotheses of the form $[P]$ and $[\bullet N]$ because such "complex" hypotheses are invertible, i.e., they can always be replaced by an equivalent set (read disjunctively) of simple contexts.

A *hypothetical judgment* $\Gamma \vdash J$ asserts some judgment J relative to the simple context Γ. In addition to assertion and denial of propositions, we can assert an entire context Δ (read conjunctively), or a contradiction #. The connection between judgments $\Gamma \vdash J$ and the typical notation for two-sided sequents "with punctuation" is a shallow syntactic transformation. Asserted propositions in Γ can be viewed as a "left context" $\Gamma_L = \{N \mid N \in \Gamma\}$, and refuted propositions as a "right context" $\Gamma_R = \{P \mid \bullet P \in \Gamma\}$. Contradiction $\Gamma \vdash \#$ corresponds to a sequent with no distinguished formula ($\Gamma_L \Rightarrow \Gamma_R$), while $\Gamma \vdash [P]$ corresponds to a right-focused sequent ($\Gamma_L \Rightarrow \Gamma_R; [P]$), $\Gamma \vdash N$ to a right-inverting sequent ($\Gamma_L \Rightarrow \Gamma_R; N$), $\Gamma \vdash \bullet P$ to a left-inverting sequent ($P; \Gamma_L \Rightarrow \Gamma_R$), and $\Gamma \vdash [\bullet N]$ a left-focused sequent ($[N]; \Gamma_L \Rightarrow \Gamma_R$). We can interpret $\Gamma \vdash \Delta$ by expansion, into a set of sequents.

The eight rules in Figure 1 describe the *canonical* ways of deriving $\Gamma \vdash J$, in the sense that any additional forms of proof can always be reduced to these rules. This is the content of Andreoli's *focusing completeness theorem.* Something that is different from Andreoli's original presentation (though similar to [2]) is that none of the rules mention any logical connectives, but only "structural connectives". The point is that the canonical rules can be stated generically with respect to a "dictionary" for the logical connectives.

Definition II.1. *A* **dictionary** *is a pair of inductively defined relations* $\Delta \Vdash [P]$ *and* $\Delta \Vdash [\bullet N]$*. It induces a* **definition**

ordering *(an abstract notion of "subformula"), taking all the propositions in* Δ *to be below* P *(resp.* N*) if* $\Delta \Vdash [P]$ *(resp.* $\Delta \Vdash [\bullet N]$*).*

Definition II.2. *In general we write* $\mathcal{D} :: \mathcal{J}$ *to indicate that* $\mathcal{D}$ *is a derivation of the judgment* $\mathcal{J}$*. A derivation* $p :: \Delta \Vdash [P]$ *is called a* **proof pattern,** *while a derivation* $d :: \Delta \Vdash [\bullet N]$ *is called a* **refutation pattern.**

The idea is that p describes the shape of a proof of P with some "holes" Δ, and likewise d describes the shape of a refutation of N with holes Δ.

Example II.3. The following inductive clauses define some of the standard polarized connectives:

$$\frac{}{\cdot \Vdash [1]} \qquad \frac{\Delta_1 \Vdash [P_1] \quad \Delta_2 \Vdash [P_2]}{\Delta_1, \Delta_2 \Vdash [P_1 \otimes P_2]}$$

$$\text{(no proof of 0)} \qquad \frac{\Delta \Vdash [P_1]}{\Delta \Vdash [P_1 \oplus P_2]} \qquad \frac{\Delta \Vdash [P_2]}{\Delta \Vdash [P_1 \oplus P_2]}$$

$$\text{(no refutation of } \top\text{)} \qquad \frac{\Delta \Vdash [\bullet N_1]}{\Delta \Vdash [\bullet N_1 \& N_2]} \qquad \frac{\Delta \Vdash [\bullet N_2]}{\Delta \Vdash [\bullet N_1 \& N_2]}$$

$$\frac{}{\cdot \Vdash [\bullet \bot]} \qquad \frac{\Delta_1 \Vdash [\bullet N_1] \quad \Delta_2 \Vdash [\bullet N_2]}{\Delta_1, \Delta_2 \Vdash [\bullet N_1 \parr N_2]}$$

$$\frac{\Delta_1 \Vdash [P] \quad \Delta_2 \Vdash [\bullet N]}{\Delta_1, \Delta_2 \Vdash [\bullet P \to N]} \qquad \frac{\Delta \Vdash [\bullet N]}{\Delta \Vdash [N^{\perp}]} \qquad \frac{}{N \Vdash [\downarrow N]} \qquad \frac{}{\bullet P \Vdash [\bullet \uparrow P]}$$

Most of these rules should be familiar from linear sequent calculus (following the above guide for translating assertion/denial into left/right positionality), but note that here we are only building *patterns,* and there is no linearity restriction on the use of hypotheses when building actual proofs and refutations. The connectives $\downarrow$ and $\uparrow$ play the important role of coercions between the two polarities, and mark leaves of patterns. □

The reader can keep this dictionary in mind to get a handle on our presentation of focusing and its relation to more standard sequent calculi—but must keep in mind that it is *only* an example, an incomplete collection of useful connectives. In general we need not make any assumptions about the dictionary, even allowing non-well-founded definition orderings (to encode recursive types).

Now, consider the canonical rules for proving or refuting a positive proposition:

$$\frac{\Delta \Vdash [P] \quad \Gamma \vdash \Delta}{\Gamma \vdash [P]} \qquad \frac{\Delta \Vdash [P] \quad \longrightarrow \quad \Gamma, \Delta \vdash \#}{\Gamma \vdash \bullet P}$$

The rule of proof follows the intuition we described for the meaning of patterns: to prove P, we must pick one of the possible patterns of proof, and fill in all of its holes. Note that the premise $\Gamma \vdash \Delta$ can be expanded, using the canonical rules, into a list of premises $\Gamma \vdash N_i$ and $\Gamma \vdash \bullet P_j$.

The rule of refutation follows the intuition we described at the beginning of this section, that continuations for positive types may be defined by pattern-matching: to refute P, it suffices to consider the shape of any possible proof of P, and thence derive a contradiction. This is a higher-order rule, and we follow the convention that any meta-variable appearing to the left of an arrow and not in the rule's conclusion is implicitly universally quantified (with tight scope). So in prose the rule is read, "[For any Γ and P,] If for all Δ such that $\Delta \Vdash [P]$, we can derive $\Gamma, \Delta \vdash \#$, then $\Gamma \vdash \bullet P$".

Finally, let's consider the last rule dealing explicitly with positive polarity, one of the rules of contradiction:

$$\frac{\bullet P \in \Gamma \quad \Gamma \vdash [P]}{\Gamma \vdash \#}$$

This says simply that one way of establishing contradiction is to prove a (positive) proposition assumed to be false.

These three rules of proof, refutation, and contradiction are of course entirely sound intuitionistically. They give a constructive interpretation of classical logic in the sense that we can consider hypothetical judgments $\Gamma \vdash \#$ as sequents $\Gamma_L \Rightarrow \Gamma_R$, and in particular as $\cdot \Rightarrow \Gamma_R$ when Γ only contains denials of positive propositions.[1]

On the other hand, the rules of proof and refutation for negative propositions do not seem so intuitionistic:

$$\frac{\Delta \Vdash [\bullet N] \quad \longrightarrow \quad \Gamma, \Delta \vdash \#}{\Gamma \vdash N} \qquad \frac{\Delta \Vdash [\bullet N] \quad \Gamma \vdash \Delta}{\Gamma \vdash [\bullet N]}$$

In particular, for negative propositions, proof means proof-by-contradiction. Negative refutation is intuitionistically *sound* (to refute N, pick a refutation pattern and fill in its holes), but in fact it is more restrictive than ordinary intuitionistic refutation. These are really "co-intuitionistic" reasoning principles. Negative polarization thus yields a more direct constructive interpretation of classical truth (cf. [43]), although it yields a less direct interpretation of classical falsehood.

One final, important point about the logical standing of the rules in Figure 1 is that the meaning of contradiction is also not complete for the intuitionistic (or co-intuitionistic) reading. In particular, we do not have *ex falso quodlibet*: $\Gamma \vdash \#$ does not let us deduce $\Gamma \vdash [P]$ (or $\Gamma \vdash [\bullet N]$). Instead, # is the contradiction of *minimal logic* [25].

Since the rules of focusing define a cut-free sequent calculus, we should expect some sort of cut-admissibility theorem. We state this as a *composition principle,* together with an *identity principle* (corresponding to admissibility of initial axioms $A \Rightarrow A$):

Principle (Composition)**.**

1) *If* $\Gamma \vdash [P]$ *and* $\Gamma \vdash \bullet P$ *then* $\Gamma \vdash \#$.
2) *If* $\Gamma \vdash N$ *and* $\Gamma \vdash [\bullet N]$ *then* $\Gamma \vdash \#$.
3) *If* $\Gamma \vdash \Delta$ *and* $\Gamma, \Delta \vdash \#$ *then* $\Gamma \vdash \#$.

Principle (Identity)**.** *If* $\Delta \subseteq \Gamma$ *then* $\Gamma \vdash \Delta$.

In fact, the demonstration of both these principles is almost trivial—the *procedures* for witnessing them can be given in an entirely generic way, for an arbitrary dictionary. Whether or not these procedures *terminate* depends on whether the definition ordering is well-founded. It is for the dictionary in Example II.3, but in general won't be, and we can accept

[1] This interpretation of classical logic is essentially Glivenko's [21].

$$\frac{p \quad \sigma}{V^+}\ p[\sigma] \qquad \frac{p \quad \to \quad E_p}{K^+}\ p{\to}E_p \qquad \frac{d \quad \to \quad E_d}{V^-}\ d{\to}E_d \qquad \frac{d \quad \sigma}{K^-}\ d[\sigma]$$

$$\frac{v \quad K^-}{E}\ K^-\$v \qquad \frac{k \quad V^+}{E}\ k\$V^+ \qquad \frac{}{\sigma}\ \cdot \qquad \frac{\sigma_1 \quad \sigma_2}{\sigma}\ (\sigma_1,\sigma_2)$$

Fig. 2. Classical canonical forms

partiality in the definition of composition, and non-well-foundedness in the derivation of identity.

B. Programs

The preceding section can already be read directly as an *intrinsic* definition [39] of a programming language. The following table describes the correspondence between logical derivations and well-typed terms, adopting standard terminology in a faithful way:

deriving	symbol	description
$\Delta \Vdash [P]$	p	value pattern
$\Delta \Vdash [\bullet N]$	d	continuation pattern
$\bullet P \in \Gamma$	k	continuation variable
$N \in \Gamma$	v	value variable
$\Gamma \vdash [P]$	V^+	value of type P
$\Gamma \vdash N$	V^-	value of type N
$\Gamma \vdash \bullet P$	K^+	continuation accepting P
$\Gamma \vdash [\bullet N]$	K^-	continuation accepting N
$\Gamma \vdash \Delta$	σ	substitution for Δ
$\Gamma \vdash \#$	E	well-typed expression

In Figure 2 we give a conceptual grammar of the language induced by focusing proofs, where all we have done is remove the precise type information from Figure 1, leaving only the symbol of the corresponding syntactic class—some concrete syntax for the conclusion is also shown to the right of each rule. For example, a rough way of restating the rules of proof and refutation is that a positive value (negative continuation) can always be decomposed as a value (cont.) pattern together with a substitution, while a positive continuation (neg. value) is determined by a map from value (cont.) patterns to expressions.

These rules only allow the construction of terms which are β-normal, η-long, and in CPS. Which fragment of CPS depends on the types one is considering, particularly their polarities. We cannot include a detailed review here, but the intuition is simple based on the two rules for forming expressions: the rule for positive polarity passes a value to a continuation variable, while the rule for negative polarity passes a continuation to a value variable (cf. [47, Ch. 4]).

There is no problem with considering additional rules for constructing terms, so long as we read them as notation for building canonical terms (either statically or dynamically). For example, given the dictionary of Example II.3, we can add these rules for building pairs:

$$\frac{\Gamma \vdash [P_1] \quad \Gamma \vdash [P_2]}{\Gamma \vdash [P_1 \otimes P_2]} \qquad \frac{\Gamma \vdash N_1 \quad \Gamma \vdash N_2}{\Gamma \vdash N_1 \& N_2}$$

The rules may look isomorphic, but they are *justified* in very different ways. To justify the $\otimes$ rule, we assume that both premises have canonical introductions, which means that we have values $V_1^+ = p_1[\sigma_1]$ and $V_2^+ = p_2[\sigma_2]$, where $p_i :: \Delta_i \Vdash [P_i]$ and $\sigma_i :: \Gamma \vdash \Delta_i$. From this we can construct $(p_1, p_2) :: \Delta_1, \Delta_2 \Vdash [P_1 \otimes P_2]$ (using the dictionary) and $(\sigma_1, \sigma_2) :: \Gamma \vdash \Delta_1, \Delta_2$, then $(V_1^+, V_2^+) :: \Gamma \vdash [P_1 \otimes P_2]$ by $(V_1^+, V_2^+) = (p_1, p_2)[\sigma_1, \sigma_2]$. Without the type annotations, we might display this local "justification" like so:

$$\begin{array}{rcl} (V_1^+, V_2^+) & = & (p_1, p_2)[\sigma_1, \sigma_2] \\ & & \text{where } V_1^+ = p_1[\sigma_1], V_2^+ = p_2[\sigma_2] \end{array}$$

To justify the $\&$ rule, we consider the conclusion's canonical *uses,* i.e., refutation patterns for $N_1 \& N_2$ with holes Δ, and show how to derive $\Gamma, \Delta \vdash \#$. Any such refutation pattern comes from one for either N_1 or N_2, with the same Δ, and in either case we can apply one of the premises to obtain $\Gamma, \Delta \vdash \#$. Inventing some reasonable notation for patterns, this reasoning could be summarized as:

$$\begin{array}{rcl} (V_1^-, V_2^-)(\pi_1; d_1) & = & V_1^-(d_1) \\ (V_1^-, V_2^-)(\pi_2; d_2) & = & V_2^-(d_2) \end{array}$$

Operationally, the different readings of these rules correspond to the fact that one builds strict pairs, the other lazy.

When writing programs, we of course also want to consider terms which are not already β-normal and η-long. Thus we can *internalize* the composition and identity principles and interpret them dynamically—in §IV-B we will examine this in a more general setting.

C. Two ways of saying the same thing

The elegance of classical polarity is that positive and negative views are perfectly symmetric. The structure of a negative value is isomorphic to the structure of a positive continuation, just as the structure of a negative continuation is isomorphic to the structure of a positive value.

Of course, this elegant symmetry can also be seen as needless redundancy, and it is tempting to define a more minimal system by cross-section. Indeed, the positive fragment of Figure 1 appears again and again (in different guises) in studies of continuation-passing, including Thielecke's CPS calculus [44], response categories [41], Jump-With-Argument [30], and tensorial logic [34]. While polarized types provide a rich language for expressing intuitions about control directly, in the classical setting there is fundamentally no loss of expressivity in limiting oneself to, say, positive types (e.g., instead of building a lazy pair, build a continuation for a strict sum, etc.).

On the other hand, as we discussed, a similar notion of polarity seems to extend to intuitionistic systems in which positive and negative fragments are *not* simply mirror images. This suggests that we try to keep our basic intuitions about polarity, but break the perfect symmetry of the classical notion. Of course, we will not do this in an arbitrary way, but one motivated by our desire to understand the laws of delimited continuations.

III. Generalized polarity

It is natural to approach delimited control by first generalizing the answer type of continuations. In classical polarized logic, every expression (i.e., the computation that results from combining a value and a continuation) has type #, or "contradiction". As we discussed, # is not contradiction in the usual mathematical sense of *the world exploded,* but rather in the sense of *something peculiar happened.* Formally, # can be interpreted as a distinguished logical atom.

So should we simply generalize # to an arbitrary proposition? Operational intuitions instead motivate that we restrict the type of answers to be positive: think that the purpose of a computation is to produce some piece of observable *data.* In any case, starting from the formal hypothesis of positive answer types, we can proceed to generalize our analysis of classical polarity in an almost mechanical way.

A. Positive answers

Generalizing contradiction, for any positive proposition P we write simply P to assert it as an *ultimate consequence*. Generalizing denial, we write $(-) \triangleright P$ for an *argument towards* P (using the funny notation to emphasize that $(-) \triangleright P$, like $\bullet(-)$, is a structural rather than a logical connective). Imagine that we now go through Figure 1 and uniformly replace # by P, and $\bullet(-)$ by $(-) \triangleright P$. What is the meaning of this formal move?

The rule of positive proof remains unchanged, while the rule of positive refutation is simply parameterized by a consequence. The twist comes on the negative side:

$$\frac{\Delta \Vdash [N \triangleright P] \quad \longrightarrow \quad \Gamma, \Delta \vdash P}{\Gamma \vdash N} \qquad \frac{\Delta \Vdash [N \triangleright P] \quad \Gamma \vdash \Delta}{\Gamma \vdash [N \triangleright P]}$$

We have implicitly updated the dictionary here with a generalized notion of refutation pattern—without being too precise yet about what that means, let us try to understand the two rules. As in the classical case, the rule of negative refutation picks a refutation pattern (now in a generalized sense), and fills in its holes. And the rule of negative proof? Following the convention we already established, since P appears to the left of the arrow and not in the rule's conclusion, it is implicitly universally quantified in the premise, i.e., "If for all P and Δ such that $\Delta \Vdash [N \triangleright P]$, we can derive $\Gamma, \Delta \vdash P$, then $\Gamma \vdash N$".

We see that negative proof is still a sort of proof-by-contradiction, but now *polymorphic over consequences.* In particular, we could almost imagine justifying it intuitionistically (bringing to mind Friedman [17]) by instantiating P with N... except that this violates polarity. In §III-B, we will explain how for a large class of negative types, a proof-by-contradiction of N can indeed be converted into an intuitionistic proof of N.

Let us clarify the new structure of the dictionary. We do not want to consider $\Delta \Vdash [N \triangleright P]$ as an arbitrary relation, but rather restrict to *consequence parametric* definitions.

Definition III.1. *We write $\alpha.\Delta$ to indicate that the* **consequence variable** *α is bound in Δ, meaning that it can appear in a hypothesis $P \triangleright \alpha \in \Delta$, and can be substituted by a positive proposition $(P \triangleright \alpha)[P'/\alpha] = P \triangleright P'$. A* **parametric dictionary** *is a pair of inductively defined relations $\Delta \Vdash [P]$ and $\alpha.\Delta \Vdash [N \triangleright -]$.*

Example III.2. The definitions of the standard negative connectives (see Example II.3) can be generalized like so:

$$\text{(no rule for } \top) \qquad \frac{\alpha.\Delta \Vdash [N_1 \triangleright -]}{\alpha.\Delta \Vdash [N_1 \& N_2 \triangleright -]} \qquad \frac{\alpha.\Delta \Vdash [N_2 \triangleright -]}{\alpha.\Delta \Vdash [N_1 \& N_2 \triangleright -]}$$

$$\frac{}{\alpha.\cdot \Vdash [\bot \triangleright -]} \qquad \frac{\alpha.\Delta_1 \Vdash [N_1 \triangleright -] \quad \alpha.\Delta_2 \Vdash [N_2 \triangleright -]}{\alpha.\Delta_1, \Delta_2 \Vdash [N_1 \parr N_2 \triangleright -]}$$

$$\frac{\Delta_1 \Vdash [P] \quad \alpha.\Delta_2 \Vdash [N \triangleright -]}{\alpha.\Delta_1, \Delta_2 \Vdash [P \to N \triangleright -]} \qquad \frac{}{\alpha.P \triangleright \alpha \Vdash [\uparrow P \triangleright -]}$$

The definitions of the standard positive connectives remain unchanged except for $N^{\perp}$, which is redefined as a more general positive connective $N \multimapdot P$:

$$\frac{\alpha.\Delta \Vdash [N \triangleright -]}{\Delta[P/\alpha] \Vdash [N \multimapdot P]} \qquad \square$$

Now the way we interpret the negative rules is as follows:

$$\frac{\alpha.\Delta \Vdash [N \triangleright -] \quad \longrightarrow \quad \Gamma, \alpha.\Delta \vdash \alpha}{\Gamma \vdash N} \qquad \frac{\alpha.\Delta \Vdash [N \triangleright -] \quad \Gamma \vdash \Delta[P/\alpha]}{\Gamma \vdash [N \triangleright P]}$$

In particular, we make explicit in the rule of negative proof that the reasoning is parametric in the type of consequence, and in the rule of negative refutation that the chosen pattern is not limited to a particular consequence.

Finally, we consider the rule for using positive denials. Under the uniform translation it becomes:

$$\frac{P \triangleright P' \in \Gamma \quad \Gamma \vdash [P]}{\Gamma \vdash P'}$$

Now, if this were the *only* rule for deriving ultimate consequences from a hypothesis $P \triangleright P'$, it would be seemingly incomplete—why shouldn't we be allowed to further analyze P' to derive something else? Instead, we will say that P' is an *intermediate consequence* (notated $.P'$),

$$\frac{P \triangleright P' \in \Gamma \quad \Gamma \vdash [P]}{\Gamma \vdash .P'}$$

and tie the knot by adding the following pair of rules:

$$\frac{\Gamma \vdash [P]}{\Gamma \vdash P} \qquad \frac{\Gamma \vdash .P \quad \Gamma \vdash P \triangleright P'}{\Gamma \vdash P'}$$

In other words, ultimate consequences can be shown directly, or derived by way of intermediate consequences. Collecting all of these remarks, we arrive at Figure 3.

Principle (Composition)**.**

1) *If $\Gamma \vdash P$ and $\Gamma \vdash P \triangleright P'$ then $\Gamma \vdash P'$.*
2) *If $\Gamma \vdash N$ and $\Gamma \vdash [N \triangleright P']$ then $\Gamma \vdash P'$.*
3) *If $\Gamma \vdash \Delta$ and $\Gamma, \Delta \vdash P$ then $\Gamma \vdash P$.*
4) *If $\Gamma \vdash P_1 \triangleright P_2$ and $\Gamma \vdash P_2 \triangleright P_3$ then $\Gamma \vdash P_1 \triangleright P_3$.*
5) *If $\Gamma \vdash [N_1 \triangleright P_2]$ and $\Gamma \vdash P_2 \triangleright P_3$ then $\Gamma \vdash [N_1 \triangleright P_3]$.*

Contexts $\Delta,\Gamma \;::=\; \cdot \mid \Delta_1,\Delta_2 \mid N \mid P \triangleright P'$

$$\frac{\Delta \Vdash [P] \quad \Gamma \vdash \Delta}{\Gamma \vdash [P]} \qquad \frac{\Delta \Vdash [P] \quad \longrightarrow \quad \Gamma,\Delta \vdash P'}{\Gamma \vdash P \triangleright P'}$$

$$\frac{\alpha.\Delta \Vdash [N \triangleright -] \quad \longrightarrow \quad \Gamma,\alpha.\Delta \vdash \alpha}{\Gamma \vdash N} \qquad \frac{\alpha.\Delta \Vdash [N \triangleright -] \quad \Gamma \vdash \Delta[P/\alpha]}{\Gamma \vdash [N \triangleright P]}$$

$$\frac{N \in \Gamma \quad \Gamma \vdash [N \triangleright P]}{\Gamma \vdash P} \qquad \frac{P \triangleright P' \in \Gamma \quad \Gamma \vdash [P]}{\Gamma \vdash .P'}$$

$$\frac{}{\Gamma \vdash \cdot} \qquad \frac{\Gamma \vdash \Delta_1 \quad \Gamma \vdash \Delta_2}{\Gamma \vdash \Delta_1,\Delta_2} \qquad \frac{\Gamma \vdash [P]}{\Gamma \vdash P} \qquad \frac{\Gamma \vdash .P \quad \Gamma \vdash P \triangleright P'}{\Gamma \vdash P'}$$

Fig. 3. Generalized rules of focusing

Principle (Identity)**.**

1) *If* $\Delta \subseteq \Gamma$ *then* $\Gamma \vdash \Delta$.
2) $\Gamma \vdash P \triangleright P$ *for all* P.

As in §II-A, the definition of the procedures for composition and identity are completely generic in the dictionary. Although the long list of composition principles may seem daunting, they are just the mutually-recursive subroutines of this procedure (which we will describe in more concrete operational terms in §IV-B).

B. Generalized ¬¬ interpretations

Here we describe how the generalized notion of polarity corresponds to a generalized class of double-negation interpretations into second-order intuitionistic logic, including different interpretations of intuitionistic logic itself. For concreteness, we limit our attention to the connectives (re)defined in Examples II.3 and III.2. We begin by defining a positive translation P^+ together with a negative (dualizing) translation $N^{-\alpha}$, from polarized formulas to second-order intuitionistic formulas free in a monadic predicate T:

$$1^+ = \mathbb{T} \quad 0^+ = \mathbb{F} \qquad \top^{-\alpha} = \mathbb{F} \quad \bot^{-\alpha} = \mathbb{T}$$

$$(P_1 \otimes P_2)^+ = {P_1}^+ \wedge {P_2}^+ \qquad (N_1 \parr N_2)^{-\alpha} = {N_1}^{-\alpha} \wedge {N_2}^{-\alpha}$$

$$(P_1 \oplus P_2)^+ = {P_1}^+ \vee {P_2}^+ \qquad (N_1 \& N_2)^{-\alpha} = {N_1}^{-\alpha} \vee {N_2}^{-\alpha}$$

$$(N -\!\!\bullet\, P)^+ = N^{-\alpha}[P^+/\alpha] \qquad (P \to N)^{-\alpha} = P^+ \wedge N^{-\alpha}$$

$$(\downarrow N)^+ = \forall\alpha.N^{-\alpha} \supset T\alpha \qquad (\uparrow P)^{-\alpha} = P^+ \supset T\alpha$$

Next we extend the translation to judgments:

$$[P]^* = P^+ \quad P^* = .P^* = TP \quad N^* = \forall\alpha.N^{-\alpha} \supset T\alpha$$

$$(P_1 \triangleright P_2)^* = {P_1}^+ \supset T{P_2}^+ \qquad ([N] \triangleright P)^* = N^{-\alpha}[P^+/\alpha]$$

$$(\cdot)^* = \mathbb{T} \quad (\Delta_1,\Delta_2)^* = \Delta_1^* \wedge \Delta_2^*$$

We also define the formulas $Unit_T = \forall\alpha.\alpha \supset T\alpha$, $Ext_T = \forall\alpha\beta.(\alpha \supset T\beta) \supset (T\alpha \supset T\beta)$, and $Mon_T = Unit_T \wedge Ext_T$, as well as the operator $I\alpha = \alpha$, observing that for any formula φ free in T, if $\vdash^{\mathrm{ip2}} Mon_T \supset \varphi$ then $\vdash^{\mathrm{ip2}} \varphi[I/T]$.

Theorem III.3. *If* $\Gamma \vdash J$ *then* $\vdash^{\mathrm{ip2}} Mon_T \supset \Gamma^* \supset J^*$

Theorem III.4. *For any positive or negative proposition* A*, if* $\vdash^{\mathrm{ip2}} A^*[I/T]$*, then* $\vdash A$.

Proof: Very similar to the proofs in the classical setting (see Theorems 3.5.1 and 3.5.2 of [47]), with the main difference that the proof of Theorem III.3 applies the assumptions $Unit_T$ and Ext_T to deal with the two new rules. ∎

The translation uses only a small fragment of second-order logic, and sometimes this means we can derive a direct correspondence with (intuitionistic) propositional logic. Let $|A|$ be the translation that collapses polarized connectives and erases coercions, i.e., with some abuse of notation:

$$|1| = |\top| = \mathbb{T} \quad |0| = |\bot| = \mathbb{F} \qquad |\downarrow| = |\uparrow| = \cdot$$

$$|\otimes| = |\&| = \wedge \quad |\oplus| = |\parr| = \vee \quad |\to| = |-\!\!\bullet| = \supset$$

Definition III.5. *A standard polarized proposition is* **orderly** *if it does not contain* $\parr$*, and* **pure** *if it does not contain* $-\!\!\bullet$*. A proposition is* **immaculate** *if it is both orderly and pure.*

Theorem III.6. *For* A *immaculate,* $\vdash A$ *iff* $\vdash^{\mathrm{ip}} |A|$

Proof: Corollary of Theorems III.3 and III.4, because we can prove in ip2 (which is conservative over ip) that $A^*[I/T] \equiv |A|$ for immaculate A. These equivalences are immediate in the case of pure positive connectives and the coercion $\downarrow$, rely on quantifier commutations in the case of orderly negative connectives, and apply a substitution of $|P|$ for α in the case $(\uparrow P)^*[I/T] = \forall\alpha.(P^* \supset \alpha) \supset \alpha$. ∎

Although there exist non-immaculate polarizations of intuitionistic logic, in general, $\parr$ is unsound with respect to intuitionistic disjunction, while $-\!\!\bullet$ is incomplete for implication. For example, it is easy to construct a closed proof of $(P \to \bot) \parr \uparrow P$ (which is isomorphic to $P \to \uparrow P$), while in general it is impossible to prove $N \& M -\!\!\bullet (\downarrow N \otimes \downarrow M)$. We note that the immaculate fragment coincides precisely with the standard definition of polarized intuitionistic logic (cf. [31], [33]), as well as with the type structure of Call-By-Push-Value [30].

But we must also make clear that although $\parr$ and $-\!\!\bullet$ may not be immaculate, they are not *immoral*: they are perfectly constructive. As the translation into second-order logic suggests, $\parr$ is essentially the Church encoding of disjunction, while (as we will explain in §IV-C) $-\!\!\bullet$ internalizes continuations as values. Both connectives are useful computationally, and because they have first-class status as logical connectives, do not interfere with the pure and orderly lives of their neighbors.

It is also worth considering the logical content of the interpretation instantiated with predicates T other than I. For example, taking $T\alpha = \#$ (the fixed atom # representing absurdity) yields classical logic as in the classical double-negation translations—except that this definition only satisfies $Unit_T$ if we explicitly *assert inconsistency* (as in [20]). With other T, we might hope to recover other modal logics and intermediate logics (e.g., $T\alpha = \alpha \vee \#$, cf. [17], [27]).

IV. Delimited control and monadic reflection

In this section we elaborate the sense in which the generalized rules of focusing provide a foundation for reasoning about delimited continuations and control operators, and their

$$\frac{p \quad \sigma}{V^+}\ p[\sigma] \qquad \frac{p \quad \to \quad E_p}{K^+}\ p{\to}E_p \qquad \frac{d \quad \to \quad E_d}{V^-}\ d{\to}E_d \qquad \frac{d \quad \sigma}{K^-}\ d[\sigma]$$

$$\frac{v \quad K^-}{E}\ K^-\$v \qquad \frac{k \quad V^+}{.E}\ k\$V^+ \qquad \frac{}{\sigma}\ \cdot \qquad \frac{\sigma_1 \quad \sigma_2}{\sigma}\ (\sigma_1,\sigma_2)$$

$$\frac{V^+}{E}\ !V^+ \qquad \frac{.E \quad K^+}{E}\ K^+\$.E$$

Fig. 4. Generalized canonical forms

interaction with side-effects. We do not have space to include many examples here, and the reader is instead referred to our Twelf formalization,[2] which closely follows and elaborates this section.

A. Syntax of normal forms

Since Figure 3 is only a slight modification of Figure 1, the interpretation of §II-B also only requires minor adjustment. Our taxonomy of values, continuations, substitutions, and expressions remains unchanged, except that continuations are associated with positive answer types, and expressions with positive types that are either synthesized or checked (depending on whether they correspond to intermediate or ultimate consequences). In Figure 4, we again give an abstract syntax of canonical forms, derived from Figure 3 in the same way Figure 2 was derived from Figure 1.

B. Dynamic semantics

Here we explain how to turn the composition and identity principles of §III-A into dynamic processes, as usual in a type-generic way. First we introduce a bit of notation:

$$1.\ K^+ \$ E \qquad 2.\ K^- \$ V^- \qquad 3.\ E[\sigma] \qquad 4.\ K_2^+ \circ K_1^+ \qquad 5.\ K_2^+ \circ K_1^-$$

Each of these constructs internalizes one of the composition principles, swapping the order of the two premises and leaving type constraints implicit. Principles (1) and (2) are justified by the following reductions to principles (3)–(5):

$$K^+\$!p[\sigma] \rightsquigarrow K^+(p)[\sigma] \qquad d[\sigma] \$ V^- \rightsquigarrow V^-(d)[\sigma]$$

$$K^+ \$ (K_1^+ \$ .E) \rightsquigarrow (K^+ \circ K_1^+) \$ .E \qquad K^+ \$ (K_1^- \$ v) \rightsquigarrow (K^+ \circ K_1^-) \$ v$$

Principles (3) and (4) are reduced as follows:

3) $E[\sigma]$ is the capture-avoiding substitution of σ into E (which creates instances of (1) and (2))
4) $K_2^+ \circ K_1^+$ is defined by $(K_2^+ \circ K_1^+)(p) = K_2^+ \$ K_1^+(p)$

Finally, (5) reduces to multiple applications of (4), relying on the following observation:

Proposition IV.1. *Any $\alpha.\Delta$ decomposes as the interleaving of a constant context Δ' with a parametric context $\alpha.P_1 \triangleright \alpha, \dots, P_n \triangleright \alpha$. Thus any substitution σ for $\Delta[P/\alpha]$ decomposes as the interleaving of a substitution σ' for Δ' together with a substitution $(K_1^+, \dots, K_n^+)$ for $P_1 \triangleright P, \dots, P_n \triangleright P$, which we express as $\sigma = \sigma' + (K_1^+, \dots, K_n^+)$.*

Composition principle (5) is therefore reduced as:

$$K_2^+ \circ d[\sigma' + (K_1'^+, \dots, K_n'^+)] \rightsquigarrow d[\sigma' + (K_2^+ \circ K_1'^+, \dots, K_2^+ \circ K_n'^+)]$$

[2] http://www.pps.jussieu.fr/~noam/delimited/

We likewise internalize the identity principles. Let $\{\Delta\}$ denote the list of variables bound by a context, and $\{p\}$ the list of variables bound by a pattern. Then the *identity substitution* $id_{\{\Delta\}}$ is defined together with the *identity continuation* Id, by the clauses $id_{\{\cdot\}} = \cdot$, $id_{\{\Delta_1,\Delta_2\}} = (id_{\{\Delta_1\}}, id_{\{\Delta_2\}})$, and

$$id_k(p) = Id \$ k \$ p[id_{\{p\}}] \qquad id_v(d) = d[id_{\{d\}}] \$ v$$

$$Id(p) = !p[id_{\{p\}}]$$

C. Delimited control operators

We have used the logical notion of polarity to reconstruct the operational notions of value, continuation, substitution and expression. Delimited control operators can already be seen as living inside this logical universe—but where? As we explained earlier, negative values are themselves a general form of control operator, defined by pattern-matching on continuations. Recall their generic introduction rule:

$$\frac{\alpha.\Delta \Vdash [N \triangleright -] \quad \longrightarrow \quad \Gamma, \alpha.\Delta \vdash \alpha}{\Gamma \vdash N}$$

The requirement of *answer type polymorphism* is a strong restriction, however, which can be relaxed by replacing values of type N with continuations of type $N \mathrel{-\!\!\bullet} P_i \triangleright P_o$ (parse this as $(N \mathrel{-\!\!\bullet} P_i) \triangleright P_o$). By definition of the positive-polarity implication $\mathrel{-\!\!\bullet}$, note the following derived inference rule:

$$\frac{\alpha.\Delta \Vdash [N \triangleright -] \quad \longrightarrow \quad \Gamma, \Delta[P_i/\alpha] \vdash P_o}{\Gamma \vdash N \mathrel{-\!\!\bullet} P_i \triangleright P_o}$$

Decomposing delimited control effects as a triple brings to mind Danvy and Filinski's original type-and-effect system [10] for shift and reset (DF89), but whereas DF89 requires various non-logical rules (including the four-place connective $\tau/\delta \to \alpha/\delta$ to represent functions with embedded control effects), we simply instantiate N with basic logical connectives. By instantiating N with $P \to N$ we derive the typing rule for functions with embedded effects, and by instantiating it with $\uparrow P$ we derive the "shift" operator, which binds a single continuation variable:

$$\frac{\Gamma, P \triangleright P_i \vdash P_o}{\Gamma \vdash \uparrow P \mathrel{-\!\!\bullet} P_i \triangleright P_o}$$

We emphasize that we are not inventing any new rules here, merely expanding definitions. A useful exercise for the reader is to likewise verify the following rules for building strict and lazy pairs of terms with delimited control effects, by the same sort of transformations we gave at the end of §II-B (we elide the fixed context Γ):

$$\frac{\uparrow P_1 \mathrel{-\!\!\bullet} P_x \triangleright P_o \quad \uparrow P_2 \mathrel{-\!\!\bullet} P_i \triangleright P_x}{\Uparrow(P_1 \otimes P_2) \mathrel{-\!\!\bullet} P_i \triangleright P_o} \qquad \frac{N_1 \mathrel{-\!\!\bullet} P_i \triangleright P_o \quad N_2 \mathrel{-\!\!\bullet} P_i \triangleright P_o}{N_1 \& N_2 \mathrel{-\!\!\bullet} P_i \triangleright P_o}$$

Observe that the answer types in the $\otimes$ rule now reflect left-to-right evaluation, whereas the unbiased & rule reflects lazy evaluation.

This analysis is very close to Kiselyov and Shan [28], who also make use of three kinds of "arrows". More or less, what we write $\to$, $\mathrel{-\!\!\bullet}$, and $\triangleright$, they write as $\to$, $\uparrow$, and $\downarrow$ (unfortunately clashing with our notation for the polarity coercions). An important feature of their type system, which is more general than DF89, is that it allows access to delimited continuations beyond the nearest dynamically-enclosing delimiter. Essentially, this is because (their) $\downarrow$ can be nested. Here, the same is achieved by *internalizing* $\triangleright$ with $\multimap\!\!\bullet$, since any positive continuation $K :: P \triangleright P_r$ can be represented as a positive value $\uparrow K :: \uparrow P \multimap\!\!\bullet P_r$. Like Kiselyov and Shan, we derive the "reset" operation, which converts $\uparrow P \multimap\!\!\bullet P \triangleright P_o$ to P_o by plugging the continuation with the value $\uparrow Id$, where Id is the identity continuation $P \triangleright P$.

Finally, we note that the relationship between values of type N and continuations of type $N \multimap\!\!\bullet P_i \triangleright P_o$ accords with Asai and Kameyama's definition [4] of *purity* as answer type polymorphism.

D. Equality and external effects

Again, we view the identity and composition principles of §IV-B as included among potentially many different *notations for canonical forms* (cf. [40]). Equivalence of notations is defined as equality of their denoted canonical forms. Now, in the classical case, we placed no non-trivial equations on focusing proofs themselves (only α-equivalence), with every syntactically distinct expression representing a semantically distinct, sequential evaluation strategy. In the general setting, though, we have new possibilities. Consider the following expression (where $.E \mathbin{⅋} K^+$ is sugar for $K^+ \$ .E$):

$$.E_1 \mathbin{⅋} p_1 \to .E_2 \mathbin{⅋} p_2 \to E_{(p_1,p_2)}$$

Assuming that p_1 and p_2 bind distinct variables and that $.E_2$ does not depend on p_1, the preceding might be considered equivalent to

$$.E_2 \mathbin{⅋} p_2 \to .E_1 \mathbin{⅋} p_1 \to E_{(p_1,p_2)}$$

On the other hand, intuitively this equivalence is only valid if $.E_1$ and $.E_2$ do not have interfering *side-effects*.

As hinted by the translation of §III-B, we can interpret proofs/programs with respect to an ambient notion of computation T, indeed a *monad* [36]. The syntactic inclusion ! of values V^+ in (checking) expressions E may be recognized as the unit of the monad, and the application of a continuation K^+ to a (synthesizing) expression $.E$ as its bind operation. In general, then, which additional equations we need to place on these canonical forms depends on the monad we have in mind. This general question is beyond our scope here, but we can consider a few interesting cases:

- T the free monad: no additional equations.
- T a commutative monad [26]: the above and similar equations should be derivable.
- $T = I$ (identity monad): further equation should be derivable, expressing that computations can be freely duplicated or thrown away.

The commutative monad equations are dealt with in CLF [45] in terms of a notion of *concurrent contexts* (which generalize evaluation contexts), and we believe this approach can be adapted to validate the additional equations of the identity monad, yielding a decision procedure in the finitary case (i.e., when each type has finitely many patterns and the definition ordering is well-founded). Combining this with Theorem III.6, it should be possible to connect equality of focusing proofs in the free/identity monad cases to normalization-by-evaluation-based decision procedures for lambda calculus with weak/strong sums [9], [1].

E. Monadic reflection

We end this section by briefly relating the concepts we've been exploring to Filinski's *monadic reflection* [14], [16], a way of mediating between the "internal view" of effects (typical of Haskell) where monads are used as a way of organizing purely functional code, and the "external view" (typical of ML) where at least a few effectful operations are built into the operational semantics of the language. Intuitively, *reflect* takes a pure piece of data of some monadic type $T\alpha$ and performs the effect it represents, yielding a computation of type α. Conversely, *reify* takes an effectful computation of type α and represents it as a pure term of type $T\alpha$. "Filinski's representation theorem" is the observation that the delimited control operators shift and reset can be used to implement reflect and reify for any monad [14].

To begin, we will explain Filinski's representation theorem in terms of elementary category theory [32].[3] We then describe how to program this theorem directly, given one more natural generalization of the proof theory of polarity.

Let Set be the category of sets. A *monad* on Set is a triple of a functor $T : \mathrm{Set} \to \mathrm{Set}$, a map $\eta_P : P \to TP$ (the *unit* of the monad) for each object $P \in \mathrm{Set}$, and a map $K^* : TP_1 \to TP_2$ (the *extension operation*) for each map $K : P_1 \to TP_2$, satisfying the equations

$$\eta_P^* = Id_{TP} \quad K^* \circ \eta = K \quad (K_1^* \circ K_2)^* = K_1^* \circ K_2^*$$

Given a monad T on Set, the *Kleisli category* Set_T shares the same objects as Set and has morphisms $\mathrm{Set}_T(P_1, P_2) = \mathrm{Set}(P_1, TP_2)$, with identity and composition defined using the monad's unit and extension operation. The functor $\mathbb{E}_T : \mathrm{Set}_T \to \mathrm{Set}$ is defined by $\mathbb{E}_T P = TP$ and $\mathbb{E}_T K = K^*$.

Now by the Yoneda lemma, for every object $P \in \mathrm{Set}_T$, there is a bijection between $\mathbb{E}_T P$ and the set of natural transformations from the hom-functor $\mathrm{Set}_T(P, -)$ to $\mathbb{E}_T$. Concretely, given an element $E \in TP$, we can form the natural transformation which maps any $K : P \to TP'$ to $K^*(E) \in TP'$. Conversely, given $V \in \mathrm{Nat}(\mathrm{Set}_T(P, -), \mathbb{E}_T)$, we can obtain $V(\eta_P) \in TP$.

We claim that the two directions of the Yoneda isomorphism *are* Filinski's original implementation of reflect and reify, up to minor details (e.g., Filinski uses a universal type to get around limitations of ML's type system).

To see this directly in polarized logic, we have to make explicit some of the type structure which has thus far been implicit in our analysis. We explained (§III-B, §IV-D) that polarized types and terms can be interpreted with respect to a monad, implicit and hence fixed. Suppose we make the monad

[3]The connection between Filinski and Yoneda has as far as I know not been previously pointed out, although a very similar connection was drawn in a blog post by Dan Piponi [37].

explicit on the negative side, defining a class of negative types N_T for any *internal monad* T. An internal monad is defined as above, by a family of continuations $\eta_P :: P \triangleright TP$ and continuation transformers $(-)^* :: P \triangleright TP' \vdash TP \triangleright TP'$, satisfying the above equations. The positive-to-negative coercion is then written $\uparrow^T P$ (annotated explicitly with an internal monad), defined by $\alpha.P \triangleright T\alpha \Vdash [\uparrow^T P \triangleright -]$. We forgo recapitulating the generic rules of inference and composition for negative types N_T, which are straightforward generalizations of the rules with the implicit monad.[4]

Now, Filinski's representation theorem is realized as a simple isomorphism between expressions of positive type TP and values of negative type $\uparrow^T P$: given an expression $E :: TP$, we can form the negative value $\uparrow k \to k^* \ \$ \ E :: \uparrow^T P$, and conversely, given a negative value $V^- :: \uparrow^T P$, we can form the expression $\uparrow \eta_P \ \$ \ V^- :: TP$.

V. Conclusion

Much of the classical literature on polarity takes place in the context of classical linear logic, with elegant connections to game semantics and continuation-passing. On the other hand, less systematic polarity-like distinctions have been invented to tackle difficult problems in intuitionistic circles, particularly to better understand effectful programming and dependent types. We have shown that "intuitionistic polarity" really does arise from the same principles as classical polarity, but in the more general framework of delimited continuation-passing, and that this setting enriches constructive logic with delimited control operators.

Although we believe this particular generalization of the notion of polarity is well-motivated, it is also clearly only a step towards better understanding and further generalization. How does our interpretation of negative polarity relate to the algebraic view in other recent studies of control and effects [24], [35]? Can the manipulations we made in §IV-E be related to the more abstract study of the Yoneda lemma [42]? Can the hierarchy of control operators [11], [5] be related to the polarized generalization of the quantifier hierarchy [7], and to hierarchies in higher category theory? And what can we learn from such relationships?

Acknowledgments. Many thanks: Andrzej Filinski, Jonas Frey, Hugo Herbelin (particularly for a thought-provoking talk [22]), Danko Ilik, Dan Licata, Paul-André Melliès, Jeffrey Sarnat, Robert Simmons, and the LICS reviewers.

References

[1] T. Altenkirch, P. Dybjer, M. Hofmann, and P. Scott. Normalization by evaluation for typed lambda calculus with coproducts. LICS 2001.
[2] J.-M. Andreoli. Focussing and proof construction. *Annals of Pure and Applied Logic*, 107(1):131–163, 2001.
[3] J.-M. Andreoli. Logic programming with focusing proofs in linear logic. *Journal of Logic and Computation*, 2(3):297–347, 1992.
[4] K. Asai and Y. Kameyama. Polymorphic delimited continuations. APLAS 2007.

[4]Intuitively, negative types N_T are functors $\mathrm{Set}_T \to \mathrm{Set}$ (answer type-indexed sets of continuations), and negative values are natural transformations $N_T \to \mathbb{E}_T$. The coercion $\uparrow^T$ represents the Yoneda embedding.

[5] M. Biernacka, D. Biernacki, and O. Danvy. An operational foundation for delimited continuations in the CPS hierarchy. *Logical Methods in Computer Science*, 2(1), 2005.
[6] K. Chaudhuri, F. Pfenning, and G. Price. A logical characterization of forward and backward chaining in the inverse method. IJCAR 2006.
[7] A. Ciabattoni, N. Galatos, and K. Terui. From axioms to analytic rules in nonclassical logics. LICS 2008.
[8] V. Danos, J.-B. Joinet, and H. Schellinx. A new deconstructive logic: Linear logic. *The Journal of Symbolic Logic*, 62(3):755–807, 1997.
[9] O. Danvy. Type-directed partial evaluation. POPL 1996.
[10] O. Danvy and A. Filinski. A functional abstraction of typed contexts. Tech. Rep. 89/12, DIKU, Univ. of Copenhagen, Denmark, 1989.
[11] O. Danvy and A. Filinski. Abstracting control. LFP 1990.
[12] R. Dyckhoff and S. Lengrand. Call-by-value λ-calculus and LJQ. *Journal of Logic and Computation*, 17:1109–1134, December 2007.
[13] M. Felleisen. The theory and practice of first-class prompts. POPL 1988.
[14] A. Filinski. Representing monads. POPL 1994.
[15] A. Filinski. *Controlling Effects*. PhD thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania, May 1996.
[16] A. Filinski. Monads in action. POPL 2010.
[17] H. M. Friedman. Classically and intuitionistically provably recursive functions. *Higher Set Theory*, 21–27. Springer-Verlag, 1978.
[18] J.-Y. Girard. A new constructive logic: Classical logic. *Math. Structures in Computer Science*, 1:255–296, 1991.
[19] J.-Y. Girard. On the unity of logic. *Annals of Pure and Applied Logic*, 59(3):201–217, 1993.
[20] J.-Y. Girard. Locus Solum: From the rules of logic to the logic of rules. *Math. Structures in Computer Science*, 11(3):301–506, 2001.
[21] V. I. Glivenko. Sur quelques points de la logique de M. Brouwer. *Bulletins de la classe des sciences*, 15:183–188, 1929.
[22] H. Herbelin. *On the logical content of control delimiters*. Talk at the PPS/πr^2 seminar *Théorie des types et réalisabilité*, Nov. 18, 2009.
[23] J. M. Howe. *Proof search issues in some non-classical logics*. PhD thesis, University of St Andrews, Dec. 1998.
[24] M. Hyland, P. B. Levy, G. D. Plotkin, and J. Power. Combining algebraic effects with continuations. *TCS*, 375(1-3):20–40, 2007.
[25] I. Johansson. Der Minimalkalkül: Ein reduzieerter intuitionistischer Formalismus. *Compositio Mathematica*, 4:119–136, 1937.
[26] D. J. King and P. Wadler. Combining monads. Glasgow Workshop on Functional Programming, 1992.
[27] O. Kiselyov. Constructive law of excluded middle. Post to TYPES mailing list, September 10, 2009.
[28] O. Kiselyov and C. Shan. A substructural type system for delimited continuations. TLCA 2007.
[29] O. Laurent and L. Regnier. About translations of classical logic into polarized linear logic. LICS 2003.
[30] P. B. Levy. *Call-By-Push-Value*. PhD thesis, Queen Mary, University of London, 2001.
[31] C. Liang and D. Miller. Focusing and polarization in intuitionistic logic. CSL 2007.
[32] S. MacLane. *Categories for the Working Mathematician* Springer-Verlag, 1971.
[33] S. McLaughlin and F. Pfenning. Imogen: Focusing the polarized inverse method for intuitionistic propositional logic. ICLP 2008.
[34] P.-A. Melliès and N. Tabareau. Resource modalities in game semantics. *Annals of Pure and Applied Logic*, 161(5):632–653, 2010.
[35] R. E. Møgelberg and A. Simpson Relational parametricity for computational effects. LICS 2007.
[36] E. Moggi. Notions of computation and monads. *Information and Computation*, 93(1):55–92, 1991.
[37] D. Piponi. Reverse engineering machines with the Yoneda lemma, Nov. 12, 2006. Weblog entry, *A Neighborhood of Infinity*.
[38] G. Restall. Multiple conclusions. In *12th International Congress on Logic, Methodology and Philosophy of Science*, pp. 189–205, 2005.
[39] J. C. Reynolds. *Theories of Programming Languages*. Cambridge University Press, Cambridge, England, 1998.
[40] H. Schwichtenberg. Finite notations for infinite terms. *Annals of Pure and Applied Logic*, 94(1–3):201–222, 1998.
[41] P. Selinger. Control categories and duality: on the categorical semantics of the lambda-mu calculus. *MSCS*, 11(2):207–260, 2001.
[42] R. Street and R. Walters. Yoneda structures on 2-categories. *Journal of Algebra*, 50:350–379, 1978.
[43] T. Streicher and B. Reus. Classical logic, continuation semantics and abstract machines. *JFP*, 8(6):543–572, Nov. 1998.
[44] H. Thielecke. *Categorical Structure of Continuation Passing Style*. PhD thesis, Laboratory for Foundations of Computer Science, University of Edinburgh, 1997.
[45] K. Watkins, I. Cervesato, F. Pfenning, and D. Walker. A concurrent logical framework I: Judgments and properties. TR CMU-CS-02-101, Dept. of CS, Carnegie Mellon Univ., 2002. Revised May 2003.
[46] N. Zeilberger. Focusing & higher-order abstract syntax. POPL 2008.
[47] N. Zeilberger. *The logical basis of evaluation order and pattern-matching*. PhD thesis, Carnegie Mellon University, Apr. 2009.